外から見た“組み込み”のしごと

伊藤昌夫

第2回 世の中は白と黒だけではない

つい最近，北京で行われたSoftware Process Workshopに出席しました．多数の招待講演のうち，米国University of MarylandのVictor R. Basili教授の以下のコメントに，「そうだよね」と共感しました．

- 航空宇宙関係では，ハードウェアについては新しいものを積極的に使おうとするが，ソフトウェアを新しくしようとはしない．
- 「コード・レビューより機能テストのほうが良い」，あるいは「機能テストよりコード・レビューのほうが有効である」といった俗説をきちんと検証しよう．

1番目のコメントは航空宇宙関係の特殊性だと思います．この分野ではコストよりも信頼性が優先されます．ですから，ハードウェアはほんとうに良いものを用います．以前に工場で見たH2ロケットのパネルは，リブを立てるのにアルミを深く削って1枚ものとしていました．LSIすら新規に起こします（使用数は当然少ない）．

その一方で，ソフトウェアもコスト度外視でどんどん新しいことに挑戦するのかというと，そんなことはありません．なるべく実績のあるコードを使って，新規のものを最小限に抑えようとします．アポロ・ロケットの時代は制御もハードウェア中心だったことから，テン・ナイン（99.99999999％）と言われる信頼性を確保できました．ソフトウェアでは，そもそも信頼性を計測することすら困難で，一般にはオーダから異なると言われています（高い品質を持つと言われるソフトウェアでも，100万行当たり数十の欠陥がある）．

2番目のコメントは，いわば俗説を数値によって検証するという話です．例えば，「コード・レビューはすべてのコードに対して実施しなくてはならない」や「機能テストはすべてのモジュールに対して網羅的に実施し，カバレッジは100％でなくてはならない」と言わないということです．もちろん，それができればよいことはわかっています．局面によっては，必要な場合もあるでしょう．しかし，納期やリソースが限られている一般のソフトウェア開発では難しくなります．

ではどうするかなのですが，思考実験でポイントは押さえられます．

● コード・レビューと機能テストには利害得失がある

コード・レビューが機能テスト（あるいは単体テスト）と比べて優れているのは，1回通しで見ることによって，複数のパスを同時にチェックできることです．テストの場合，テスト・ケースごとに，たった一つのパスしかチェックすることができません．また，コーディング・スタイルにおける一貫性のような機械では判定しづらいものも，人間の目を通すことでチェックできます．さらに，経験の浅いレビューワへの教育的な効果もあります．

欠点は，人間系に頼っていることです．経験豊かなプログラマが簡単に見つけられる誤りを，そうでないプログラマは見過ごすかもしれません．

テストが優れているのは，次の点です．動的なふるまいは，コードを追うだけでは見つけにくいものです．時系列のコードは良いのですが，例えば複数のタスクのリソース競合といったことは，にわかにはわからないと思います．実際にプログラムを動かして確認するということで，多少の安心を得ることができます．また，テスト・ケースそのもののレビューを行うことで，テスト・ケースを客観的に評価できるということもあります．かりに経験の少ないプログラマが不十分なテスト・ケースを作ったとしても，それをレビューで指摘できます．コード・レビューの場合はそれができません．唯一，チェック・リストなどで個人差を緩和するしかありません．

欠点は明らかです．あらゆる入力ベクトルをテストすることはできないということです．せいぜい，コード・カバレッジ（全コード中でテストが通過した割合）やパス・カバレッジ（すべての分岐のうち，テストが通過した割合）といった基準を設けるしか手はなく，それでもすべての入力値を試せるわけではありません．

● 状況に合わせて柔軟に適用する

さて，良い解決策は明らかだろうと思います．「状況に応じて使い分ける」．先の航空宇宙分野のケースもその状況の一つの例です．もし，開発者が経験豊かならば，逐次コード・レビューを行っていくべきです．そうでないならば，重要なコードから順次リソースと納期を見極めてコード・レビューを行う．テストについても同様です．

それでは，回答になっていない？ 白でもない，黒でもないとしたら，どの程度の灰色にすればよいのかがわからない？ そうかもしれません．Basili教授が考案したクリーンルーム手法では，形式的な方法を適用することにより，統計的に信頼性を実現できるようにしています．ただし，コストがかかります．くふうを通してなんらかの形で簡便にソフトウェアの信頼性を表現することが可能となれば，ハードウェアよりも信頼される日が来るかもしれません．そうなれば，効率の良い開発を行えるようになります．また，出荷後，ドキドキする日々を送らなくてすむのではないかと思います．

<筆者プロフィール>
伊藤昌夫（いとう・まさお）．自動車会社，航空機関連会社のソフトウェア・エンジニアを経て，ニルソフトウェアを設立．ソフトウェア・プロセスおよび開発環境が専門で，そのためのコンサルテーションおよびツールの提供を行っている．設計における人間の認知活動に興味を持っている．